## Table of contents

### Chapter-1

### Centroid tracking using image sensor data



### Chapter-2

### Target identity estimation using image and acoustic sensor data

## Centroid Tracking and Target Identity Estimation using Image Sensor Data

Automatic detection and tracking of targets using imaging sensor data is essential in air traffic control and air defense applications. In this report, two important aspects: centroid tracking and target identity estimation using image sensor data are covered in two chapters 1 and 2.

### Chapter -1
### Centroid tracking using image sensor data

### 1.1 Introduction

In image-based air traffic control or air defense system, automatic detection and tracking of targets are extremely important for their safety or early warning. In such scenario, the sensor images are often cluttered, dim, spurious or noisy due to the fact that the distances to targets from the control center are large. Tracking problems involve processing measurements from a target of interest and producing at each time step, an estimate of the target's current position and velocity vectors. Uncertainties in the target motion and in the measured values, usually modeled as additive random noise, lead to corresponding uncertainties in the target state. Also, there is additional uncertainty regarding the origin of the received data, which may or may not include measurements from the targets and may be due to random clutter (false alarms). This leads to the problem of data association [1]. In this situation tracking algorithms have to include information on detection and false alarm probabilities. A comparison of the commonly used algorithms for data association and tracking namely Nearest Neighbour Kalman filter (NNKF) and Probabilistic data association filter (PDAF) is made in ref. [2] for single target tracking in clutter. In this Chapter, the results of performance of the two data association algorithms for the centroid-tracking problem are presented.

The main focus in this chapter is the implementation and validation of the algorithm for precision tracking with segmentation from imaging sensors [3]. The algorithm termed "Image Centroid Tracking algorithm "(ICTA) is independent of the size of targets and less sensitive to the intensity of the targets. Typical characteristics of the target obtained by motion recognition or by object (pattern) recognition methods are used in associating images to the target being tracked. Motion recognition characteristics of a target are its location, velocity

and acceleration, which are obtained using data from successive frames (inter-scan level). Object (pattern) recognition characteristics are its geometric structure (shape, size), energy level distribution (i.e. different gray level in the image) in one or more spectral bands, which is computed at the intra scan level.

The ICTA combines both object and motion recognition characteristics for practical target tracking from imaging sensors. The characteristics of the image considered are the intensity and size of the cluster. The pixel intensity is discretized into several layers of gray level intensities and it is assumed that sufficient target pixel intensities are within the limits of certain target layers. The ICTA implementation involves the conversion of the image into a binary image and applying upper and lower threshold limits for the "target layers". The binary target image is then converted to clusters by using nearest neighbor criterion. If the target size is known, then it is used to set limits for removing those clusters that differ sufficiently from the size of the target cluster to reduce computational complexity. The centroid of the clusters is then calculated and this information is used for tracking the target. The ICTA involves the following steps:

i. Identifying potential targets by image segmentation methods
ii. Calculation of centroid of the identified targets
iii. Tracking centroid using single or multiple target tracking techniques
iv. Separation of the true and false targets by gating and data association techniques based on both motion and object characteristics

In this chapter of the report, the results of tracking the centroid of single and multiple synthetic images in clutter is presented. The tracking performance of ICTA is evaluated in terms of Percentage-Fit-Error (PFE), Root Mean Square Position Error (RMSPE), and Root Mean Velocity Error (RSSVE). Also results of state vector fusion of data from a ground based radar and imaging sensor are presented.

### 1.2 Segmentation and Centroid Technique

Segmentation means decomposition of the image under study into its different areas of interest. In ICTA algorithm, particle segmentation is used to separate the target (object of interest) from background, when target is not fully visible [2]. It is assumed that the pixel intensities are discretized into 256 gray levels. The particle segmentation can be done in

two steps: 1) The gray level image is transformed into binary image using lower and upper threshold limits of the target. These thresholds of target can be determined through the pixel intensity histograms from the target and its surroundings. 2) The pixels which fall within this target layer limits (called pixel detections) are computed along with the probability of detection $p(\mathrm{i,j})$ (pixel detection probability of pixel $(\mathrm{i,j})$). The detected pixels are grouped into clusters with nearest neighbor technique [3].

The gray image $\mathrm{I(i,j)}$ is converted into binary image with the intensity $\beta(\mathrm{i,j})$:

$$\beta(\mathrm{i,j}) = \begin{cases} 1 & \mathrm{I_L} \le \mathrm{I(i,j)} \le \mathrm{I_U} \\ 0 & \text{otherwise} \end{cases} \tag{1.1}$$

where $\mathrm{I_L}$ and $\mathrm{I_U}$ are the lower and upper threshold limits of the target.

The detection probability of the pixel $(\mathrm{i,j})$ can be defined as:

$$\begin{aligned} P\{\beta(\mathrm{i,j}) = 1\} &= p(\mathrm{i,j}) \\ P\{\beta(\mathrm{i,j}) = 0\} &= 1 - p(\mathrm{i,j}) \end{aligned} \tag{1.2}$$

where $p(\mathrm{i,j}) = \dfrac{1}{\sigma\sqrt{2\pi}} \int_{\mathrm{I_L}}^{\mathrm{I_U}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}\, dx$, considering the gray image $\mathrm{I(i,j)}$ as Gaussian with mean $\mu$ and variance $\sigma^2$.

The binary image is then grouped into clusters using the nearest neighbor technique. A pixel is considered as belonging to the cluster only if the distance between this pixel and at least one other pixel of the cluster is less than the proximity distance $d_p$. The $d_p$ should be such that:

$$\sqrt{\frac{1}{p_t}} < d_p < \sqrt{\frac{1}{p_v}} \tag{1.3}$$

where $p_t$ and $p_v$ are detection probabilities of target and noise pixels respectively.

The centroid of the cluster is determined using:

$$x_{n_c} = \frac{\sum_{k=1}^{N} x_{n_k} I_k}{\sum_{k=1}^{N} I_k} \tag{1.4}$$

where $x_{n_k}$ is the $n^{\text{th}}$ co-ordinate of pixel $k$, $N$ is the number of pixels in the cluster, $k$ is the pixel index and $I_k$ is $k^{\text{th}}$ pixel intensity.

## 1.3 Data Simulation, Results and Discussions

### *1.3.1 Target Motion Models and Data Simulation*

The mathematical basis for generation of synthetic image [4] is briefly described below:

Consider two-dimensional array of

$$m = m_{\xi} \times m_{\eta} \tag{1.5}$$

pixels where each pixel is represented by a single index $i = 1, \cdots, m.$ and the intensity $I$ of pixel $i$ is given by

$$I_i = s_i + n_i \tag{1.6}$$

where, $s_i$ is the target intensity and $n_i$ is the noise intensity in pixel $i$. The image noise is modeled as being Independent and Identically Distributed (IID) with zero mean and covariance $\sigma^2$.

The total target-related intensity is given by:

$$s = \sum_{i=1}^{m} s_i \tag{1.7}$$

If the number of pixels covered by the target is denoted by $m_s$, then the average target intensity over its extent is given by:

$$\mu_s = \frac{s}{m_s} \tag{1.8}$$

and the average target intensity over entire frame is given by

$$\bar{s} = \frac{s}{m} \tag{1.9}$$

The average pixel SNR (over the extent of the target) is

$$r' = \frac{\mu_s}{\sigma} = r\frac{\sqrt{m}}{m_s}, \quad \text{where } r \text{ is frame SNR} \tag{1.10}$$

From eqs. (1.5-1.10) it is clear that the synthetic images (rectangular or circular) in a frame can be generated by using the following inputs:

(i) Target pixel intensity: $N\left(\mu_t, \sigma_t^{\,2}\right)$
(ii) Noise pixel intensity: $N\left(\mu_n, \sigma_n^{\,2}\right)$
(iii) Target Type: 1-'Rectangle', 2-'Circle'
(iv) Target dimension: Base (NX) & Height (NY) or radius for circular target
(v) Position of a target in each scan: x-position and y-position

In order to simulate the motion of the target in the image frame, the kinematic models of target motion are used. Both constant velocity and constant acceleration models [1] are

considered for generation of the data, which determines the position of the target in each scan.

#### *1.3.1.1 State Model*

The following state and measurement models are used to describe the constant velocity target motion.

State model:
$$X(k+1)=\begin{bmatrix}1 & T & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & T\\ 0 & 0 & 0 & 1\end{bmatrix}X(k)+\begin{bmatrix}\frac{T^2}{2} & 0\\ T & 0\\ 0 & \frac{T^2}{2}\\ 0 & T\end{bmatrix}w(k) \qquad (1.11)$$

where state $X(k)=[x_{pos.}\ \mathrm{x}_{\mathrm{vel.}}\ y_{pos.}\ \mathrm{y}_{\mathrm{vel.}}]$, $T$ =sampling period and $w(k)$ is zero mean Gaussian process noise with variance $Q$.

Measurement model:
$$z(k+1)=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}X(k+1)+v(k+1) \qquad (1.12)$$

where $v(k)$ is the centroid measurement noise that is zero mean Gaussian noise with variance:
$$R=\begin{bmatrix}\sigma_x^2 & 0\\ 0 & \sigma_y^2\end{bmatrix} \qquad (1.13)$$

Process noise and centroid measurement noise are assumed to be uncorrelated.

#### *1.3.1.2 Clutter Model*

In a practical scenario, since the image measurements could have clutter caused by interference from other targets and lower resolution due to limited spatial coverage of the imaging sensor. It is assumed that the clutter has a Poisson distribution for purposes of simulation. Assuming that a sensor has N resolution cells (pixels), detection is declared in a cell if the output in a pixel exceeds a certain threshold. Due to sensor noise or background noise, the sensor may give detections even though the sensor points to a region where there are no targets. The probability mass function for m cells in the volume V is given by
$$\mu_{FA}(m)=e^{-\lambda V}\frac{(\lambda V)^m}{m!} \qquad (1.14)$$

where $\lambda$ is the spatial density $\lambda = \frac{N\, p_{FA}}{V}$ (1.15)

where $p_{FA}$ is false alarm probability, $V$ is the volume of $N$ cells under consideration.

### *1.3.2 Tracking performance using ICTA*

Once the image data is generated, at each instant of time from the image data, the ICTA algorithm determines the centroid of the target using segmentation technique [3]. As described in section-1.2, the segmentation technique divides the image into three layers of gray level intensities. The target is assumed to be in one of the layers. A two-dimensional array of $64 \times 64$ pixels is considered for the image. The target, a two-dimensional array of $m = n_x \times n_y$ pixels, is modeled as white Gaussian random field with a mean $\mu_t$ and variance $\sigma_t^2$. The image is converted into binary image using the upper and lower limits of a target layer as threshold limits ($I_L$ and $I_H$), and then grouped into clusters by the nearest neighbor technique using the optimal proximity distance $d_p$. The centroid of each cluster is calculated and used for state estimation in the measurement update part of the NNKF or PDAF filters to track the target in clutter. The NNKF and PDAF algorithms include track initiation and track deletion features, which is essential in multi target tracking in clutter [1]. The ICTA algorithm has been validated with a single target data. It was found that the performance of both the NNKF and PDAF filters were similar for tracking the centroid of the target in clutter. All the results are evaluated by performing 25 Monte Carlo simulations.

The ICTA algorithm has also been used for tracking two crossing targets in clutter. Here at each instant of time, the centroids of both the targets are determined from the imaging sensor data. For two targets tracking it was observed that the NNKF tracking performance was not acceptable whereas the PDAF was found to track both the targets in clutter. The performance of the two algorithms is evaluated in terms of [5]:

i. The percentage fit error (PFE) in x and y

$$PFE_x = 100 * \frac{norm(x - \hat{x})}{norm(x)} \qquad PFE_y = 100 * \frac{norm(y - \hat{y})}{norm(y)} \tag{1.16}$$

ii. The root mean square position error

$$RMSPE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \frac{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2}{2}} \tag{1.17}$$

iii. The root mean square velocity error

$$RMSVE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{(\dot{x}_i - \hat{\dot{x}}_i)^2 + (\dot{y}_i - \hat{\dot{y}}_i)^2}{2}} \quad (1.18)$$

iv. The root sum square position error

$$RSSPE = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} \quad (1.19)$$

v. The root sum square velocity error

$$RSSVE = \sqrt{(\dot{x} - \hat{\dot{x}})^2 + (\dot{y} - \hat{\dot{y}})^2} \quad (1.20)$$

where $x$ and $y$ are the measurements, $\hat{x}$ and $\hat{y}$ are the estimated (initiated) target locations in x and y coordinates, respectively.

Fig. 1.1 shows the frame, which includes the estimated and true data of two crossing targets in clutter. The frame shows the background clutter, the two synthetic target images at 50$^{th}$ scan along with the true and the estimated trajectory using PDAF algorithm for tracking. The frame also shows the gate formed around the target track at each scan for both the targets. Figs. 1.2a and 1.2b show estimated position and velocities compared with the true values. The state errors with their bounds and RSSPE and RSSVE values for the two targets are shown in Figs. 1.3a and 1.3b. Percentage fit error in x- & y-positions and root mean square position error for two targets are within the acceptable range as shown in Table-1.1. State errors and root sum square errors in position and velocities using twenty-five Monte Carlo simulations are shown in Fig. 1.4. The state errors are with in the theoretical bounds and the root sum square errors are fraction of a pixel, which shows the robustness of the algorithm.

### 1.4 Track-to-Track Fusion

Having validated the performance of the ICTA for multiple target tracking, the algorithm is used for providing input to state vector fusion when the data of position from ground based radars is available in Cartesian coordinate frame. Flow diagram for fusion of data from imaging sensor and ground based radar is shown in Fig. 1.5. The two state vectors are fused using the following relations [6]:

Consider two tracks whose state vector estimates $(\hat{x})$ and covariance matrices $(\hat{P})$ are given at scan k:

$$\text{Track } i: \hat{x}_i(k), \hat{P}_i(k)$$

$$\text{Track } j: \hat{x}_j(k), \hat{P}_j(k) \quad (1.21)$$

The combined/fused estimate is given by

$$\hat{x}_c(k) = \hat{x}_i(k \mid k) + \hat{P}_i(k \mid k)\hat{P}_{ij}(k)^{-1} \left[\hat{x}_j(k \mid k) - \hat{x}_i(k \mid k)\right] \quad (1.22)$$

The combined covariance matrix associated with the estimate of (1.22) are given by

$$\hat{P}_c(k) = \hat{P}_i(k \setminus k) - \hat{P}_i(k \mid k)\hat{P}_{ij}(k)^{-1}\, \hat{P}_i(k \mid k) \quad (1.23)$$

where $\hat{P}_{ij}$ is cross covariance between $\hat{x}_i(k \mid k)$ and $\hat{x}_j(k \mid k)$, and is given by

$$\hat{P}_{ij}(k) = \hat{P}_i(k \mid k) + \hat{P}_j(k \mid k) \quad (1.24)$$

The true, estimated and fused trajectories are shown in Fig. 1.6 from which it is clear that the fused trajectory matches the true trajectory.

## 1.5 Concluding remarks

Image Centroid Tracking Algorithm (ICTA) has been developed in PC MATLAB for accurate target tracking based on the data obtained from imaging sensors, when the target is not fully visible during tracking. Using segmentation technique the gray level image has been converted into binary image and reduced into cluster by nearest neighbor (NN) criteria. Two tracking filters, namely Nearest Neighborhood Kalman Filter and Probabilistic Data Association Filter) were employed for state estimation using centroid measurement of clusters. The simulation results show that it is possible to achieve tracking accuracies of 0.6 pixels root mean square error in position and 0.2 pixels/frame in velocity. Simulation results validate the performance prediction of the proposed algorithm. ICTA has been extended to multi-sensor scenario, where the trajectories obtained from both imaging sensor and ground based radar are fused using track-to-track fusion technique.

## 1.6 References


1) Y. Bar Shalom and T.E. Fortmann. “Tracking and Data Assciation”, Academic Press, New York, 1988.
2) Girija G. and Raol J.R., Comparison of methods for association and fusion of multi-sensor data for tracking applications. AIAA Guidance, Navigation and Control Conference & Exhibit, Montreal, Quebec, Canada, Paper No. AIAA-2001-4106, 06-09, August 2001.
3) E.Oron, A. Kumar, and Y. Bar-Shalom, “Precision Tracking with segmentation for imaging sensor ”, *IEEE Transaction on Aeroapce and Electronics Systems,* AES-29, pp.977-987, July-1993.
4) A.Kumar, Y.Bar-Shalom, and E.Oram, “Precision tracking based on segmentation with optimal layering for imaging sensor”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol-17, pp 182-188,July 1995.
5) V. P. S. Naidu, Girija G. and J. R. Raol, “Evaluation of data association and fusion algorithms for tracking in the presence of measurement loss”, *AIAA Conference on Navigation, Guidance and Control*, Austin, USA, 11-14 August 2003.

6) K.C.Chang, R.K Saha, Y,Bar-Shalom, " On Optimal Track-to-Track Fusion", *IEEE Transaction on Aerospace and Electronics Systems,* AES-33**,** pp.1271-1276, Oct 1997.

**Table-1.1 Percentage fit error and root mean square error of both trcak1 and track2**

| | PFEx | PFEy | RMSPE |
|---|---|---|---|
| **Track1** | 1.34 | 1.41 | 0.52 |
| **Track2** | 2.19 | 3.39 | 0.92 |

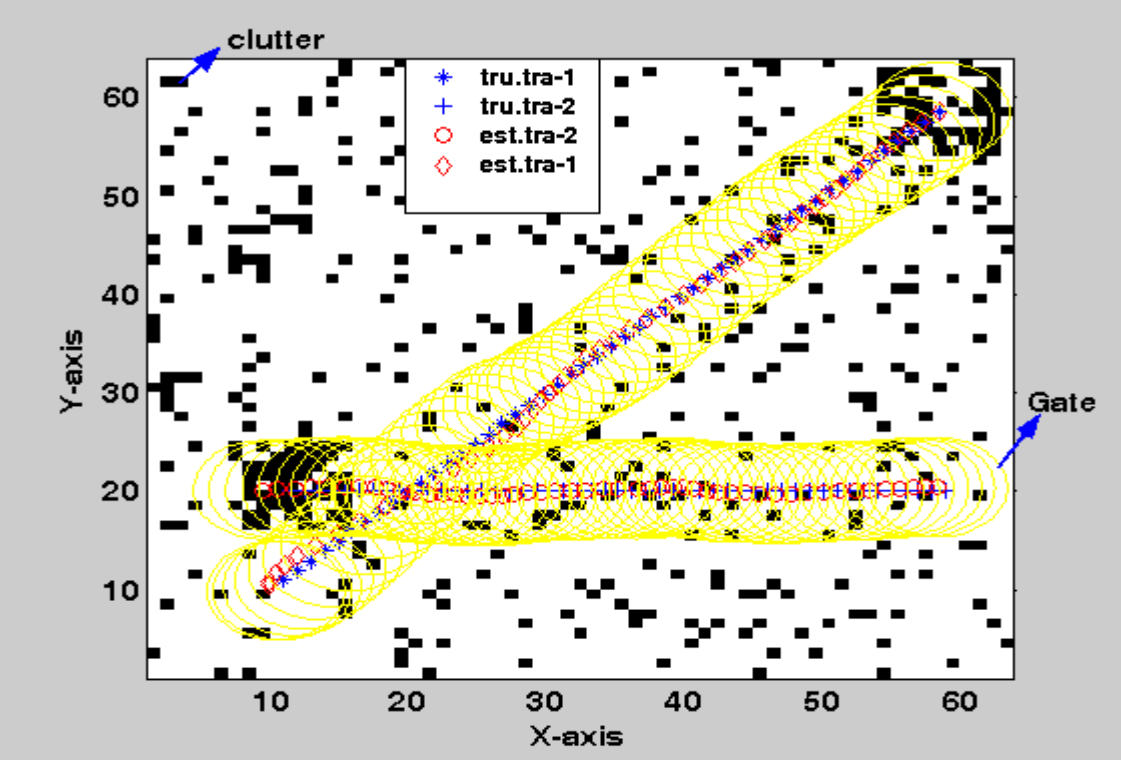


**Fig. 1.1 Tracking of two Targets in presence of clutter**

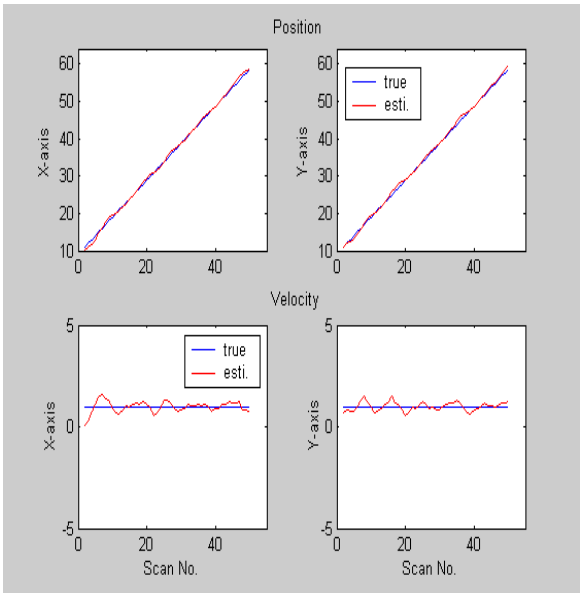


**Fig. 1.2a True and estimated positions and velocities for track1**

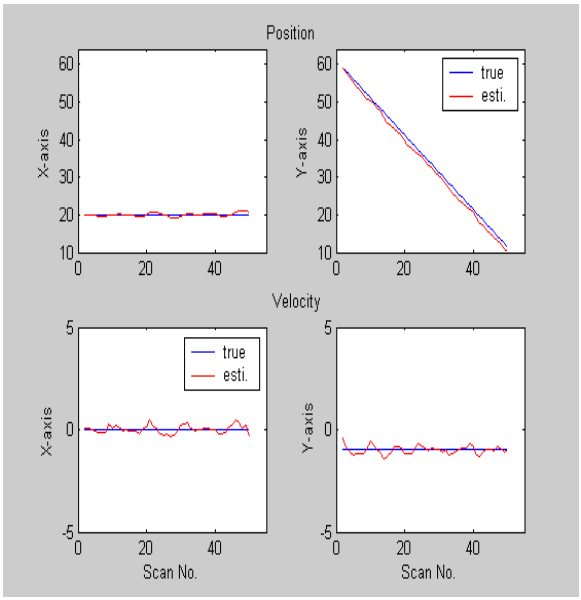


**Fig. 1.2b True and estimated positions and velocities for track2**

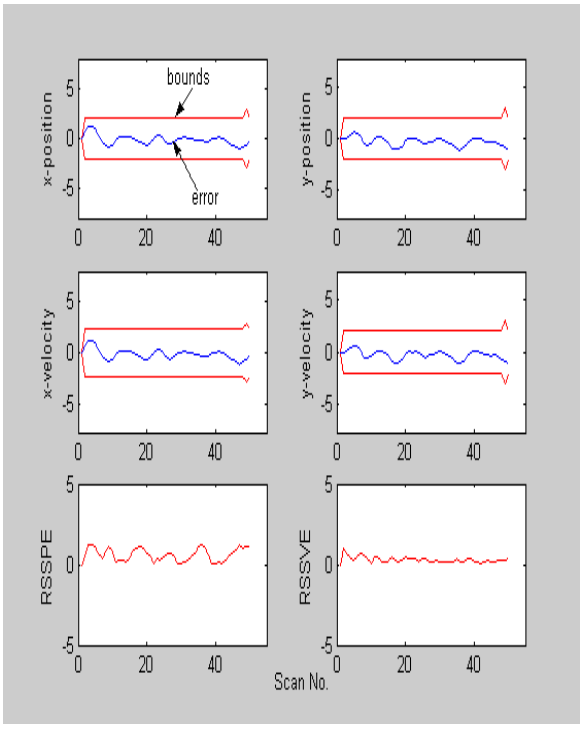


**Fig. 1.3a State errors in positions and velocities, RSSPE and RSSVE for track1**

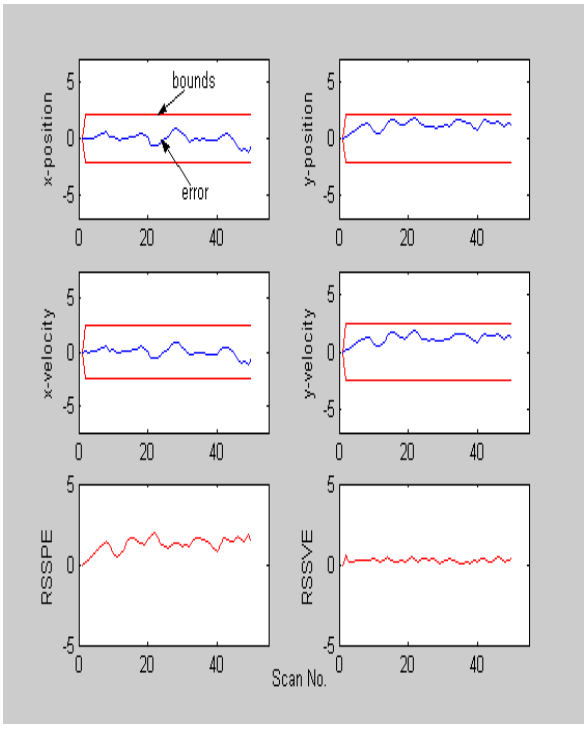


**Fig. 1.3b Residuals in positions and velocities RSSPE and RSSVE for track2**

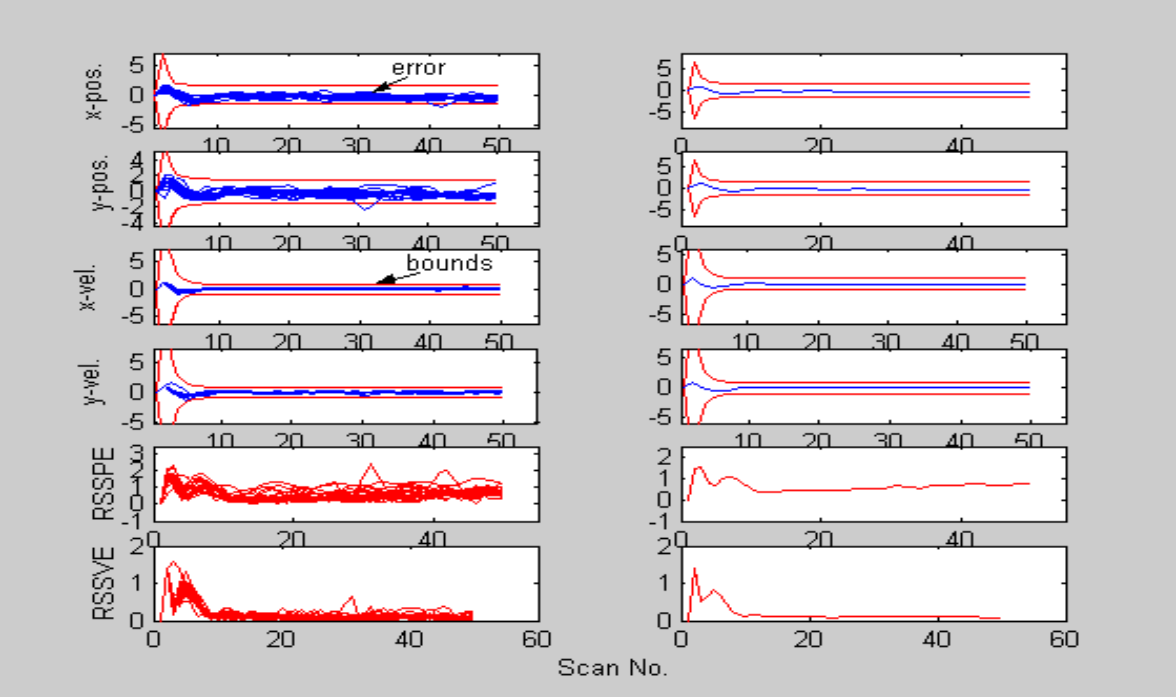


**Fig. 1.4 Sate errors and root sum square errors in positions and velocities using twenty-five Monte Carlo simulations**

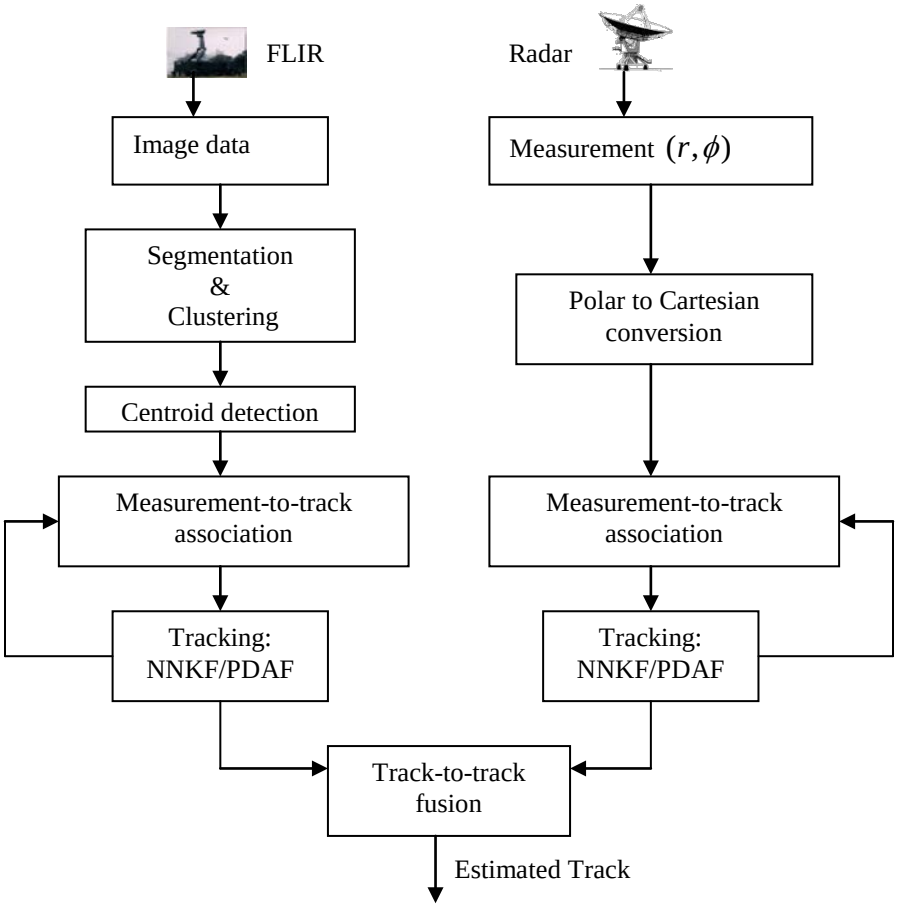


**Fig. 1.5 Flow diagram for fusion of data from imaging sensor and ground based radar**

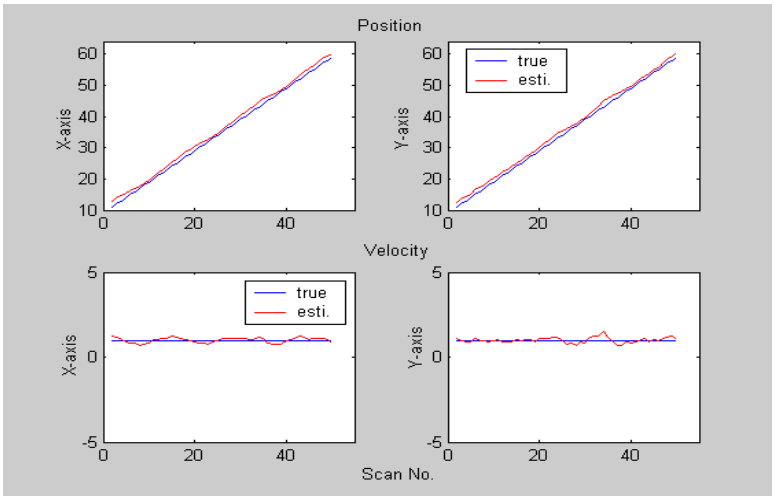


**Fig. 1.6a True and estimated positions and velocities from ground based radar**

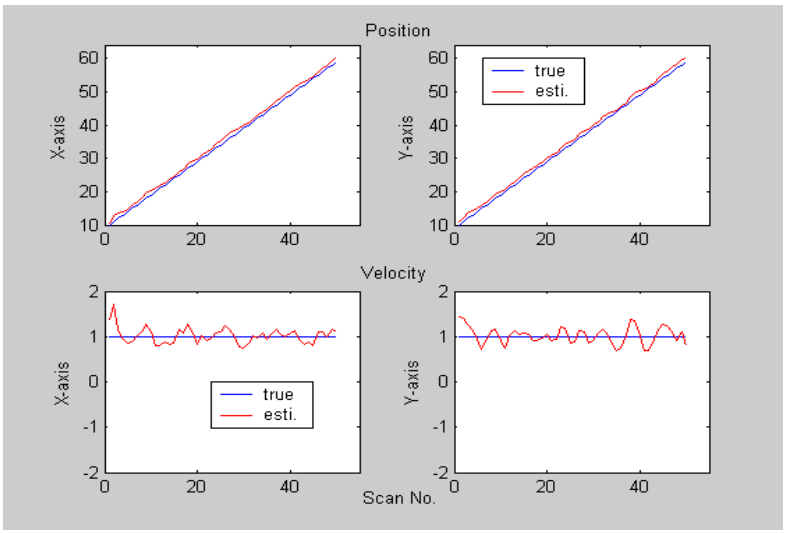


**Fig. 1.6b True and estimated positions and velocities from imaging sensor**

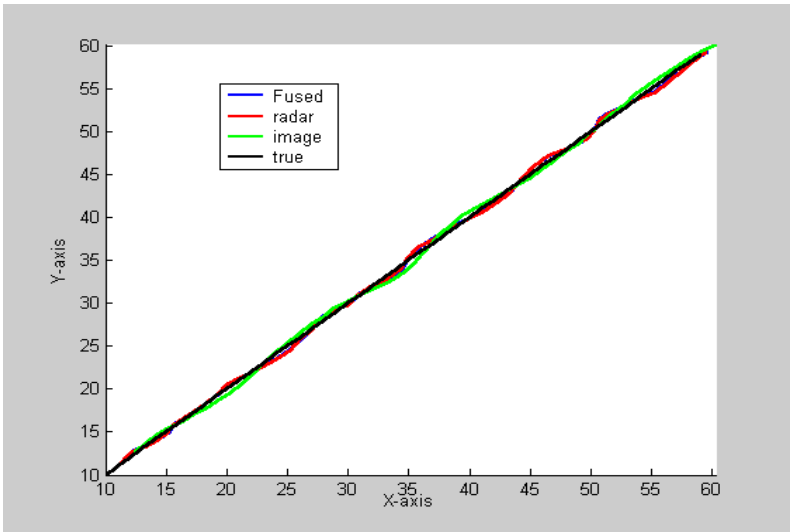


**Fig. 1.6c Fused trajectory**

## Chapter-2

# Target identity estimation using image and acoustic sensor data

### 2.1 Introduction

Automatic target recognition (ATR) [1] is one of the key components of present and future defense weapon systems to be used on autonomous vehicle missions. The key advantage of ATR is that it removes man from the process and this makes the task of recognition more efficient, reliable and robust. One important application of the ATR is in helping and guiding pilots of high-performance aircraft flying close to the ground during bad weather or at night. Examples of systems incorporating ATR are the low altitude navigation and targeting infrared for night (LANTIRN) system, cruise missile and remotely piloted vehicle (RPV) applications.

In this Chapter, a technique for estimation of target location and identity using simulated data of an infrared and acoustic sensors is presented. Infrared sensor detects all the targets in its FOV and generates images, which provide information of both the target location and identity. Acoustic sensor provides data, which helps to determine the direction of the target. These sensor outputs or their extracted features cannot be fused directly since the information from the two sensors is disparate. Bayesian fusion [2] enables fusion of such data by fusing the posterior probabilities of the sensor outputs. In this method, inference about the scene, i.e. the location and identity of objects is obtained by

i) Formulating a prior distribution for the scene
ii) Constructing probability models for multiple-sensor data conditioned on the scene
iii) Generating unified inference about the scene using the posterior distribution of the scene for given sensor data

Probabilities of the respective sensors are obtained as the product of the current likelihood and prior probability of the estimate of the scene. The method assumes that there exists a space of all the possible estimates where every estimate has a corresponding posterior probability. In this case, the posterior probability of a given estimate is the fused posterior probability. Thus the fused data of the sensors corresponding to all the possible estimates form the posterior distribution.

The ultimate goal is to obtain the optimal estimate or an approximation to an optimal estimate. Therefore, the actual problem involves obtaining an estimate with maximum

posterior probability from the posterior distribution. This is done by using search algorithms. In this Chapter, a comparison of the performance of the following three search algorithms for optimal estimation of location and identity of targets is presented.

**Metropolis Hastings (MH):** This algorithm simulates the posterior distribution by generating samples from the proposed distribution given in ref. [3]. Metropolis-Hastings algorithm is a powerful MCMC (Markov Chain Monte Carlo) method to simulate multivariate distributions. Distributions are simulated by generating Markov chain. This algorithm proceeds iteratively accepting and rejecting states $X_s^t$ in the space $\chi_s$. Once the desired posterior distribution is reached, the mode of the distribution is reported as the solution. The correctness of the result depends on how well the distribution is simulated. The desired result is the state with maximum posterior probability of the actual distribution. Thus, this algorithm could be viewed as a search algorithm looking for the state with maximum posterior distribution of the actual distribution, by simulating the distribution.

**Simulated Annealing (SA):** This algorithm is based on the simulation of the process of annealing (the way the hot gasses are cooled slowly to obtain glass) [4]. Simulated annealing is a method for the global optimization problem and it is a generic probalistic/heuristic approach. This method is inspired from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The atoms become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy due to heat. Finally, slow cooling gives them more chances of finding configurations with lower internal energy than the initial one. The function to be minimized has the form $F(X) = -\log(P'(X))$ where $P'$ is the posterior distribution and the transition probability which decides the acceptance of a generated state is based on Boltzmann acceptance criteria.

The algorithm implementation involves two nested loops. The inner loop works with the temperature specified by the outer loop. Initially the temperature is high and the transition probability of accepting states with higher energy (the lower posterior probability) is high. As the temperature decreases the probability of higher energy states being accepted reduces. Temperature is reduced according to a chosen schedule. At lower temperature simulated annealing is similar to greedy algorithm where only the lower energy states are accepted. At the end of simulation the state corresponding to last iteration is reported as solution.

**Gradual Greedy (GG):** This method is a combination of principles of simulated annealing and Metropolis Hastings algorithm. The simulated annealing algorithm proceeds according to some chosen schedule, which decides how the temperature is reduced. As the temperature reduces the probability of higher energy states being accepted reduces. In other words it could be said, as temperature reduces the greediness increases, i.e. the algorithm becomes more and more greedy gradually. This algorithm is a modified version of MH algorithm where the modification involves introducing greediness while accepting the states.

The states are generated using the proposed distribution of ref. [3] as in MH algorithm with a specified transition probability. The given state is accepted if its transition probability is greater than or equal to acceptance threshold. Initially the threshold is set to zero, at every iteration the threshold is increased and the algorithm ends when threshold equals one. Thus like SA algorithm, initially the states with lower posterior probability are accepted when the acceptance threshold is low, but as the acceptance threshold increases, the chances of accepting states with lower posterior probability decreases and finally the state with higher posterior probability is obtained.

The methodology for target location and identity estimation using IR and acoustic sensor data is validated using simulated data of the sensors. The report presents the mathematical models used for synthesizing the scenario data required for simulation of sensor data as well as models of the sensors. A program has been developed on PC MATLAB for data simulation and estimation. Fig. 2.1 shows the block/information flow diagram for target identity and location estimation using Bayesian fusion and search algorithms.

### 2.2 Data Synthesis

Data synthesis for comparison of the algorithms uses realistic models for simulating the IR and acoustic sensor data. Generation of data from the sensors requires creation of a scenario where there are several targets placed at different locations and directions. This scenario forms the input to the sensor mathematical models for generating the data. In the current work for simplicity, the scene $X_s$ is considered to be two-dimensional. Further, the targets are assumed to be at fixed orientations and of the same size. The same techniques could be applied to targets with different orientations and sizes.

The mathematical models for both infrared and acoustic sensors need some input scenario to generate the data. In case of infrared sensors, the input is the radiations emitted from the targets present in the field of view (FOV). Therefore the synthesized data, which will be input to the mathematical model of infrared sensor, is an image consisting of targets placed at random positions. The intensity of the image pixel will then represent the radiation from the targets in the FOV. For acoustic sensor, the input is the acoustic signals from the targets present in the FOV. Since acoustic sensor can detect only the direction of targets, the synthesized data for a mathematical model of acoustic sensor is an 'n' component vector consisting of directions only.

The input to the sensors is synthesized by using the concept of Marked Spatial Point Process (MSP). If 'S' denotes the continuum space, then the realization of a random point process on 'S' is a set of points having coordinates in 'S'. In a marked point process [5,6] each point in the set of points is a realization of point process that is associated with some auxiliary variable. The continuum space could be one, two (area) or three dimensional (volume). When the space considered is a domain of area or volume, the marked point process becomes an MSP process. For scenario simulation, auxiliary variables associated with the points are considered as the locations and identities of the targets. Four different elements $(X_{s1}, X_{s2}, X_{s3}, X_{s4})$ generated by MSP process are shown in Fig. 2.2. In this case, the MSP is used to generate a 2 dimensional scenario with 2 targets. In the matrices $X_{s1}$ to $X_{s4}$ each column represents a target with the first 2 rows giving the x, y coordinates of the target location and the third row the identity of the target. It can be observed that in each case the positions, identities and number of targets vary in a random manner. The synthesized scenario data for the space $X_s$ is shown in Fig. 2.3.

The IR sensor generates the output images by convolving the original image object space (i.e. the simulated scenario) with a Point Spread function (PSF) which is a non negative space dependent function and multiplying it by the probability that the quantum of optical energy incident on the $j^{th}$ pixel is converted into an electron in that pixel. Random noise with a known standard deviation is added to the generated image to account for the noise that is present on the on-chip amplifier through which all pixel values are read. Fig. 2.4 shows the output of the IR sensor for a typical scenario along with the PSF. Noise g(j) is added to this output to get the realistic image.

For acoustic sensor the synthesized data is a ' n' component vector of directions

$$\text{with } \theta_i = \tan^{-1}\left(\frac{a_y - y_i}{a_x - x_i}\right) \tag{2.1}$$

where $y_i, x_i \in q_i$ is the position of the i[th] target and $(a_y, a_x)$ is the location of acoustic sensor. Fig. 2.5 shows the synthesized data for acoustic sensor model with the location of acoustic sensor at $(a_y, a_x)$ = (150, 0). The simulated scene (synthesized data) consists of 3 targets along the directions of 84°, 45° and 45°. The direction of targets is with respect to acoustic sensor which is assumed to be at the bottom of Fig. 2.6 on, which the signals from the target shown with white lines strike. The response of acoustic sensor which is obtained by plotting $\| A(t) - d(\theta) \|$ for $\theta$ ranging from 1° to 90°, is also shown in the figure. Here $A(t)$ denotes the acoustic signal at time $t$ and $d(\theta)$ denotes the direction vector at an angle $\theta^0$. It is observed that the response of acoustic sensor shows peak along the directions of the target, since the three targets are present along two directions $45^0$ and $84^0$, two main peaks can be seen along those directions.

### 2.3 Bayesian fusion

Bayesian sensor fusion methodology combines the IR sensor and acoustic sensor outputs by fusing their posterior probabilities. The posterior probabilities for the sensor outputs are their likelihood functions. Applying Bayes' rule and assuming conditional independence of the sensor outputs $Y_i$'s for a given X, the posterior probability is given by:

$$P(X \mid Y_1,\ldots,Y_p) \propto L_1(Y_1 \mid X)\ldots L_p(Y_p \mid X)\, P(X) \tag{2.2}$$

where $P(X \mid Y_1,\ldots,Y_p) \equiv P'$ is Posterior distribution and $P(X)$ is apriori distribution.

Thus, the Bayesian fusion involves only multiplying the posterior probabilities of the respective sensors.

### 2.4 Estimation of target identity and location

After obtaining the a posteriori probability distribution using Bayesian fusion, the target identity and location are obtained by searching for appropriate state $X_s$ in the state

space $\chi_s$. In this report, the performance of three search algorithms: MH, SA and GG are evaluated for target location and identity estimation for a typical scene shown in Fig. 2.7.

Fig. 2.8 shows the plot of posterior energy for MH, SA and GG algorithms for 1000 iterations with input being the fused Infrared and Acoustic sensors data. It can be seen that the posterior energy ultimately decreases in all the 3 cases, which indicates that the algorithms tend towards the solution. The posterior energy is computed using

$$PE = -\log(Posterior\ \ probability) \tag{2.3}$$

For performance comparison of the three algorithms, the following parameters are evaluated.

a. Success rate $(sr)$
b. X Position Error $(xpe)$ :
c. Y Position Error (*ype*)
d. Id error $(ide)$ : Number of targets misclassified
e. Target count error $(tce)$

The best $(bt)$, worst $(wt)$ and average $(avg)$ values of the parameters 'b' to 'e' are obtained by executing the algorithm for 25 runs. The desired $bt$, $wt$, $avg$ values for all these parameters are zeros. The RSE (root square error) for the '*sr'* is computed using

$$RSE_{sr} = \sqrt{(d_{sr} - o_{sr})^2} \tag{2.4}$$

The RSSE (root sum squares error) for each of the parameters *xpe, ype, ide and tce* is computed using

$$RSSE_{bt} = \sqrt{(d_{bt.xpe} - o_{bt.xpe})^2 + (d_{bt.ype} - o_{bt.ypr})^2 + (d_{bt.ide} - o_{bt.ide})^2 + (d_{bt.tce} - o_{bt.tce})^2} \tag{2.5}$$

Similarly $RSSE_{wt}$ and $RSSE_{avg}$ are computed for *wt* and *avg* values.

where $d$ : Desired value of parameter

$o$ : Estimated value of parameter and subscript refers to parameter.

The resultant root sum square error $(RRSSE)$ is obtained from the $RSSE$ of each parameter as: $RRSSE_{bt} = \sqrt{\dfrac{\sum_{i=1}^{4} RSSE_{bt}(i)}{4}}$ , $i = 1$ to $4$ means scenario 1 to 4 (2.6)

Similar computations are carried out for $RRSSE_{wt}$, $RRSSE_{avg}$ and $RRSSE_{od}$.

For comparing the algorithms, the state matrices shown in Fig. 2.7 are used for the generation of simulated data. Tables 2.1 to 2.4 and Figs 2.9 to 2.10 give a comparison of the different parameters 'a' to 'e' for the three search algorithms. From Table 2.1 and Fig. 2.9, which show the success rates of the three algorithms, it is clear that GG has the maximum success rate especially for Scenario-IV. From Table 2.4 and Fig. 2.10 which show the $RRSSE$ for the three algorithms, the GG algorithm shows the lowest $RRSSE_{avg}$ value.

## 2.5 Concluding remarks

This chapter presents a scheme for estimation of target identity and location using synthesized scenario data. A PC MATLAB program has been developed which helps in scenario generation using marked spatial point, infrared red and acoustic data generation using appropriate models. Bayesian sensor fusion is achieved by computing the likelihood functions for the two sensor data and fusing the posterior probabilities. From the posterior probability distribution, the target identity and location are established using search algorithms. A comparison of three search algorithms is made for a two dimensional scenario with two targets. Comparative performance is studied based on several criteria. It is seen that Gradual Greedy algorithm has a slightly better performance than Metropolis Hasting and Simulated annealing algorithms.

## 2.6 References


1) Bir Bhanu, "Automatic Target Recognition: State of the Art Survey", IEEE transactions on aerospace and electronics systems, vol. AES-22, No.4 july 1996.
2) Michael. J. Smith "Bayesian Sensor Fusion a framework for using multi-modal sensors to estimate target location and identities in a battlefield scene" PhD thesis, Florida state University, 2003.
3) A. Lanterman, M. Miller, and D. Snyder," General Metropolis-Hastings jump diffusions for automatic target recognition in infrared scenes", Optical engineering, 36(4), pp 1123–1137, 1997.
4) Elaine rich & Kevin knight, "Artificial Intelligence", second edition 1991, Tata Mcgraw-Hill publishing company limited, New Delhi.
5) Xavier Descombes and Josiane Zerubia,, "Marked Point Process in Image Analysis", IEEE Signal Processing Magazine, pp 77-84, September 2002.
6) D.L Synder , "Random Point Processes", John Wiley & Sons, New York, 1995 .

**Table 2.1 Success rate of different algorithms for different inputs (for a total of 25 runs)**

| Algorithm | Scenario-I | | | Scenario-II | | | Scenario-III | | | Scenario-IV | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Su | Fa | Sr | Su | Fa | Sr | Su | Fa | Sr | Su | Fa | Sr |
| MH | 24 | 1 | 96 | 25 | 0 | 100 | 25 | 0 | 100 | 6 | 19 | 24 |
| SA | 25 | 0 | 100 | 23 | 2 | 92 | 22 | 3 | 88 | 21 | 4 | 84 |
| GG | 24 | 1 | 96 | 25 | 0 | 100 | 25 | 0 | 100 | 25 | 0 | 100 |

Su: Successes, Fa: Failures, Sr: Success rate (%)

**Table 2.2** $bt$ , $wt$ , $avg$ **values for parameters b to e**

| Sc | Al | *xpe* | | | *ype* | | | *ide* | | | *tce* | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *bt* | *wt* | *avg* | *bt* | *wt* | *avg* | *bt* | *wt* | *avg* | *bt* | *wt* | *avg* |
| I | MH | 0 | 10 | 0.4 | 0 | 12 | 0.48 | 0 | 1 | 0.04 | 0 | 0 | 0 |
| | SA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | GG | 0 | 4 | 0.16 | 0 | 20 | 0.8 | 0 | 1 | 0.04 | 0 | 0 | 0 |
| II | MH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SA | 0 | 15 | 1.16 | 0 | 17 | 1.28 | 0 | 0 | 0 | 0 | 0 | 0 |
| | GG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| III | MH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SA | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.12 |
| | GG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| IV | MH | 0 | 2 | 0.88 | 0 | 2 | 0.2 | 0 | 0 | 0 | 0 | 0 | 0 |
| | SA | 0 | 3 | 0.24 | 0 | 1 | 0.04 | 0 | 0 | 0 | 0 | 0 | 0 |
| | GG | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Sc: Scenario, Al: Algorithm

**Table 2.3** $RSSE$ **values for parameters b to e**

| | Scenario-I | | | Scenario-II | | | Scenario-III | | | Scenario-IV | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MH | SA | GG | MH | SA | GG | MH | SA | GG | MH | SA | GG |
| $RSSE_{bt}$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $RSSE_{wt}$ | 15.65 | 0 | 20.4 | 0 | 22.67 | 0 | 0 | 1 | 0 | 2.83 | 3.16 | 0 |
| $RSSE_{avg}$ | 0.63 | 0.82 | 0 | 0 | 1.73 | 0 | 0 | 0.12 | 0 | 0.91 | 0.24 | 0 |

**Table 2.4 RRSSE of different types of input**

| Algorithm | $RRSSE_{bt}$ | $RRSSE_{wt}$ | $RRSSE_{avg}$ |
|---|---|---|---|
| MH | 0 | 7.95 | 0.55 |
| SA | 0 | 11.46 | 0.96 |
| GG | 0 | 10 | 0.41 |

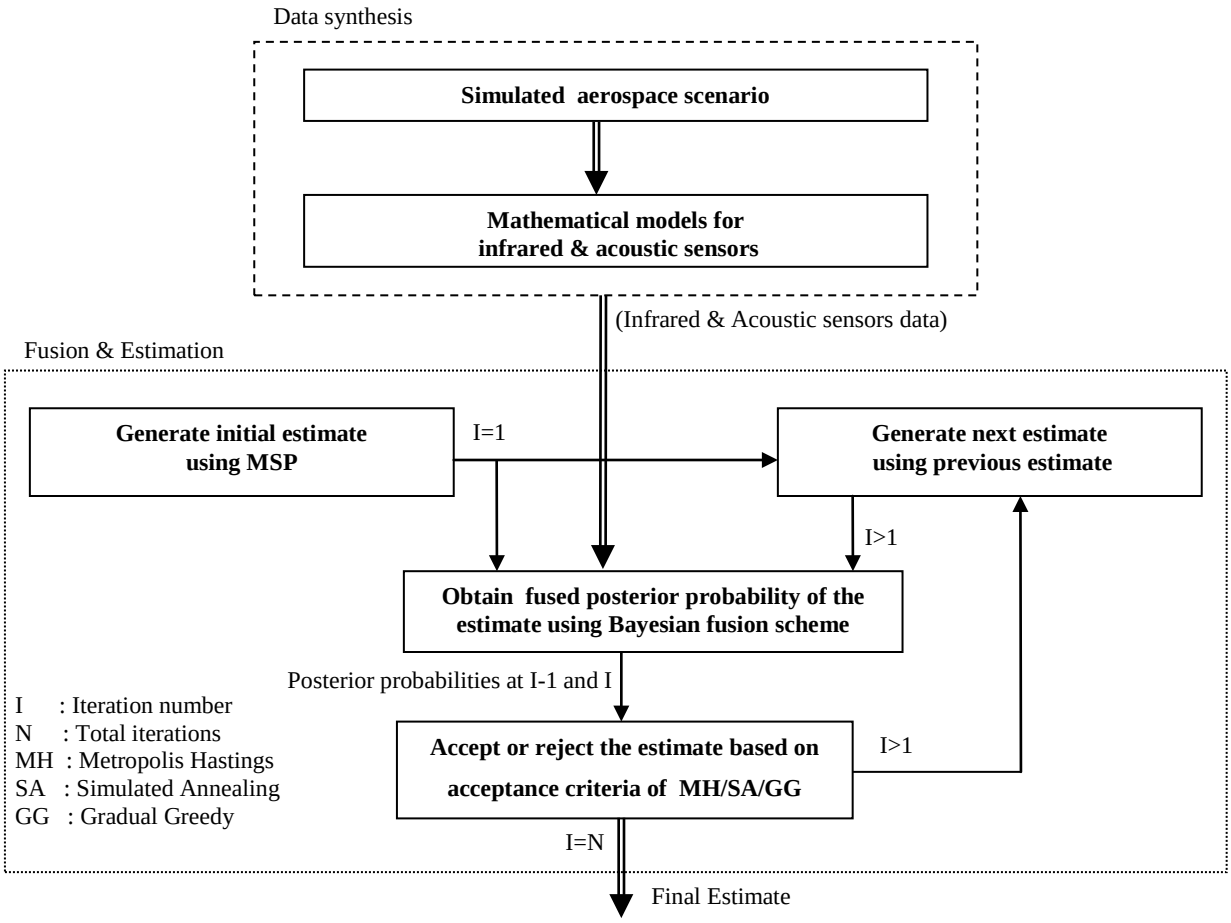


**Fig. 2.1 Block diagram for target identity and location estimation using Bayesian fusion**

$$X_{s1} = \begin{bmatrix} 37 & 82 \\ 84 & 91 \\ 1 & 2 \end{bmatrix}; \qquad X_{s2} = \begin{bmatrix} 21 & 94 & 34 & 64 & 98 & 55 \\ 107 & 67 & 70 & 21 & 107 & 105 \\ 1 & 2 & 1 & 1 & 1 & 2 \end{bmatrix};$$

$$X_{s3} = \begin{bmatrix} 32 & 108 & 54 & 21 \\ 30 & 51 & 104 & 87 \\ 1 & 2 & 1 & 1 \end{bmatrix}; \qquad X_{s4} = \begin{bmatrix} 94 & 43 & 75 & 22 \\ 106 & 101 & 66 & 63 \\ 2 & 1 & 2 & 1 \end{bmatrix};$$

**Fig. 2.2 Elements $X_s$ generated by MSP process**

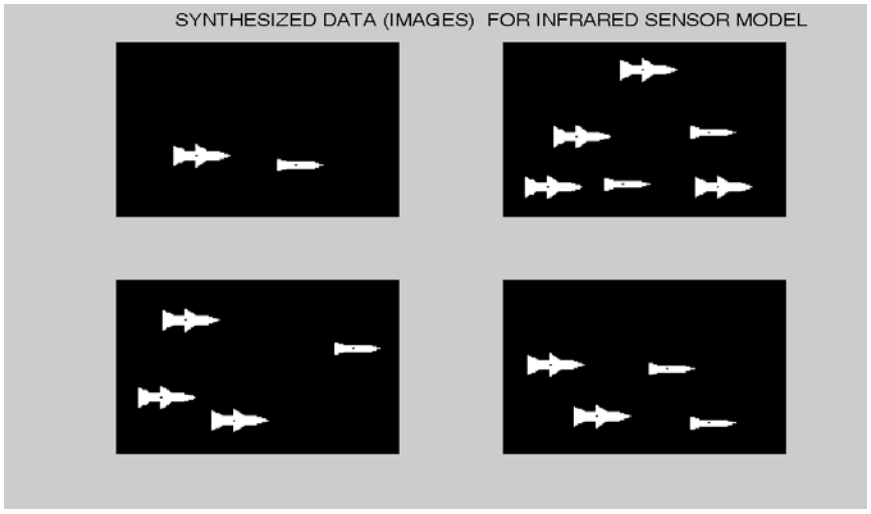


**Fig. 2.3 Synthesized images generated using the elements $X_s$**

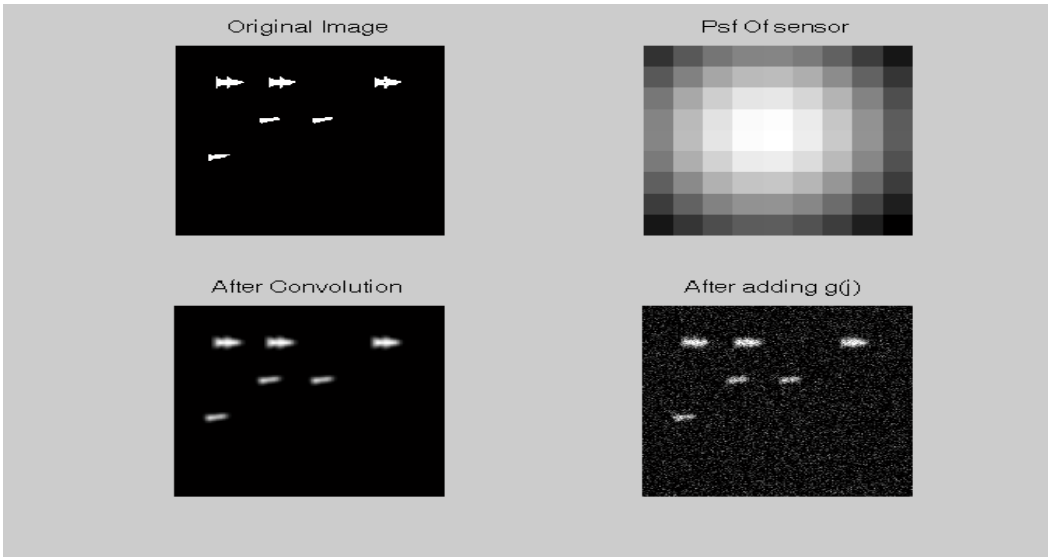


**Fig. 2.4 Generation of infrared image form the model**

$$\theta_{s1} = \begin{bmatrix} 60.72^o & 35.74^o \end{bmatrix}$$

$$\theta_{s2} = \begin{bmatrix} 63.97^o & 41.44^o & 66.97^o & 63.6^o & 23.69^o & 39.29^o \end{bmatrix}$$

$$\theta_{s3} = \begin{bmatrix} 25.08^o & 48.73^o & 48.24^o & 75.81^o \end{bmatrix}$$

$$\theta_{s4} = \begin{bmatrix} 75.07^o & 42.51^o & 40.43^o & 71.57^o \end{bmatrix}$$

**Fig. 2.5 Acoustic sensor model synthesized data generated using elements $X_s$**

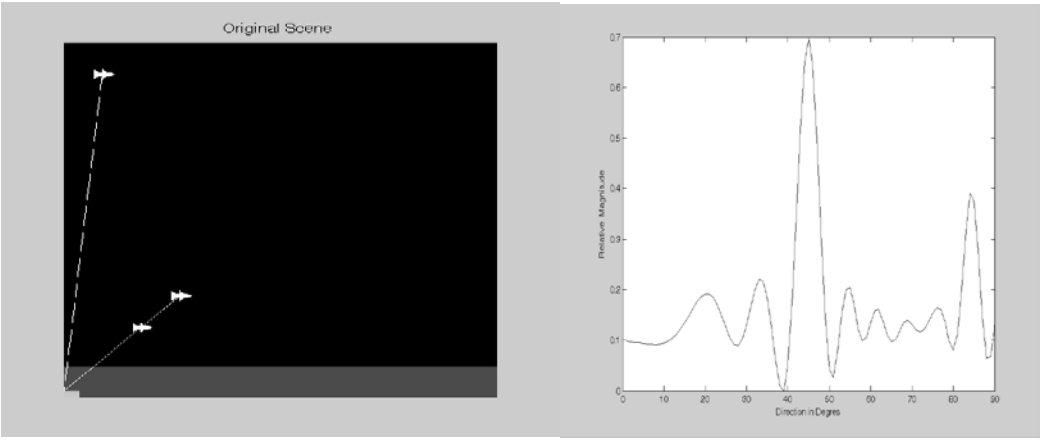


**Fig. 2.6 Simulated scene and corresponding response of acoustic sensor**

*Only Aeroplane*

$$Scenario-I=\begin{bmatrix}30\\63\\1\end{bmatrix}$$

*Aeroplane & Missile*

$$Scenario-III=\begin{bmatrix}44 & 23\\27 & 82\\2 & 1\end{bmatrix}$$

*Only Missile*

$$Scenario-II=\begin{bmatrix}94\\51\\2\end{bmatrix}$$

*Miscellaneous*

$$Scenario-IV=\begin{bmatrix}56 & 33 & 74 & 22 & 96\\103 & 38 & 44 & 86 & 96\\1 & 2 & 2 & 1 & 1\end{bmatrix}$$

**Fig. 2.7 State matrices used for generating inputs**

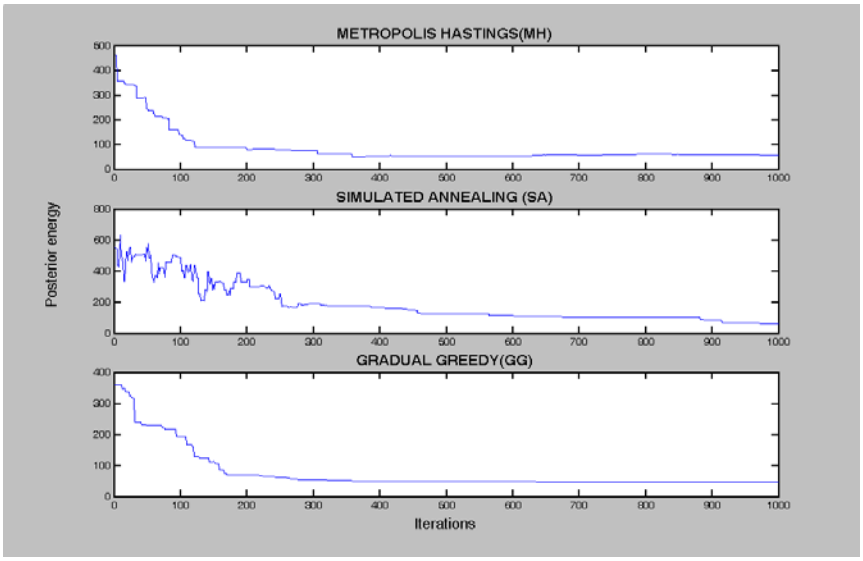


**Fig. 2.8 Posterior energy convergence for the MH, SA and GG algorithms**

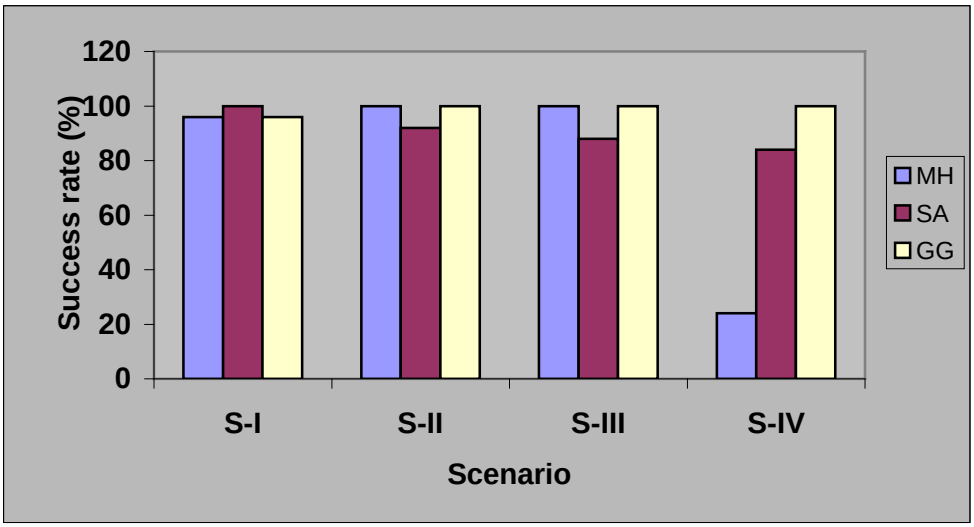


**Fig. 2.9 Success rate of different algorithms for different scenarios**

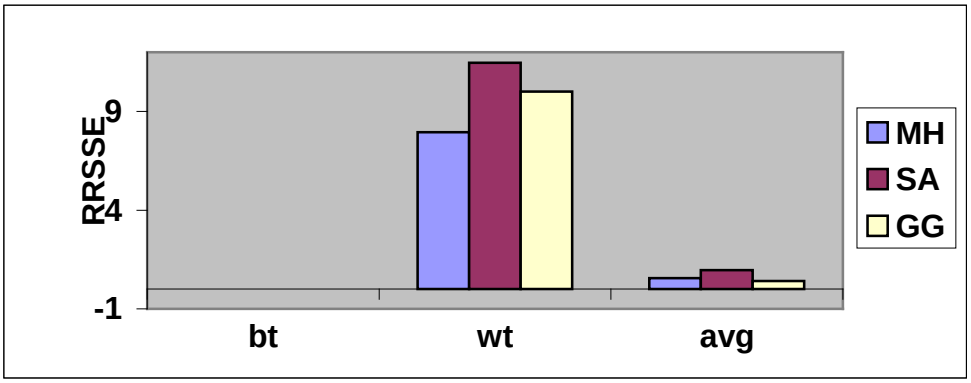


**Fig. 2.10 RRSSE for different types of input scenarios**

| | | |
|---|---|---|
| | National Aerospace Laboratories | Classification OPEN<br>No. of copies 8 |
| Title | **Centroid Tracking and Target Identity Estimation using Image Sensor Data** | |
| Author/s | V. P. S. Naidu, Girija. G and J. R. Raol | |
| Division | FMCD | NAL Project No. I-888-1/7 |
| Document No. | PD FC 0503 | Date of issue 16 Mar. 2005 |
| Contents | Pages 26 Figures 16 Tables 5 References 12 | |
| External Participation | ------ | |
| Sponsor | | |
| Approval | Head, FMCD | Group Leader |
| Remarks | | |
| Keywords | Data Fusion, Tracking performance, Image Fusion Centroid tracking, Image processing | |


Abstract

Algorithms for centroid tracking and target identity estimation using image sensor data are implemented in PC MATLAB. For tracking using image sensor data, the centroid of images is determined by using segmentation technique. This is achieved by converting the gray level image into binary image and reduced to clusters by nearest neighbor (NN) criterion. The performance of the algorithm for centroid tracking has been evaluated using simulated data.

Bayesian sensor fusion technique is used for estimation of target identity and location using simulated data of an imaging sensor and an acoustic sensor. This is achieved by computing the likelihood functions for the two sensor data and fusing the posterior probabilities. From the posterior probability distribution, the target identity and location are established using search algorithms. A comparison of three search algorithms: Metropolis Hastings, Simulated Annealing and Gradual Greedy is made for a two dimensional scenario with two targets.